(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2002－342186
（P2002－342186A）
(43)公開日　平成14年11月29日(2002.11.29)

(51)Int.Cl.⁷	識別記号	Ｆ Ｉ		テーマコード（参考）
Ｇ０６Ｆ　13/00	３５７	Ｇ０６Ｆ　13/00	３５７Ａ	５Ｂ０８９
Ｈ０４Ｌ　12/28	１００	Ｈ０４Ｌ　12/28	１００Ｈ	５Ｋ０３３

審査請求　未請求　請求項の数16　ＯＬ　（全 12 頁）

(21)出願番号　特願2002－70837(P2002－70837)

(22)出願日　平成14年３月14日(2002.3.14)

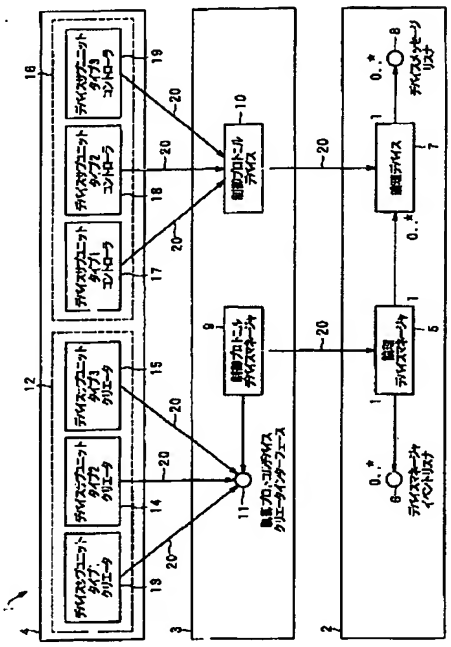
(31)優先権主張番号　０１１０６５９７．６
(32)優先日　平成13年３月15日(2001.3.15)
(33)優先権主張国　欧州特許庁（ＥＰ）

(71)出願人　598094506
ソニー　インターナショナル　（ヨーロッパ）　ゲゼルシャフト　ミット　ベシュレンクテル　ハフツング
ドイツ連邦共和国　10/85　ベルリン　ケンパープラッツ　1
(74)代理人　10006/736
弁理士　小池　晃　（外２名）

最終頁に続く

(54)【発明の名称】　機能モジュール及びホームネットワーク機器制御方法

(57)【要約】
【課題】　必要とする演算リソースが少なく、容易に拡張できる簡潔なアーキテクチャを実現する。
【解決手段】　ホームネットワーク機器に関するイベントを登録し、イベントをホームネットワーク機器に送信する機能を提供する管理ユニットと、適応化手段及びデバイス制御手段を有するデバイスユニットとを備え、デバイス制御手段は、ホームネットワーク機器固有の機能を提供し、ホームネットワーク機器を制御し、適応化手段により適応化されて、ホームネットワーク機器に関するイベントによりホームネットワーク機器の状態を変更する。



【特許請求の範囲】

【請求項１】　ホームネットワーク機器の制御ユニットにより使用される機能モジュールにおいて、

ホームネットワークにおけるホームネットワーク機器に関するイベントを登録し、該イベントを上記ホームネットワーク機器に送信する機能を提供する管理ユニットと、

上記管理ユニットに接続され、適応化手段及びデバイス制御手段を有するデバイスユニットとを備え、

上記デバイス制御手段は、ホームネットワーク機器固有の機能を提供し、該ホームネットワーク機器を制御し、

上記適応化手段により適応化されて、上記ホームネットワーク機器に関するイベントによりホームネットワーク機器の状態を変更することを特徴とする機能モジュール。

【請求項２】　上記管理ユニットは、汎用ユニット及び制御プロトコルユニットを備え、

上記汎用ユニットは、上記制御プロトコルユニット及び上記デバイスユニットに共通の機能を提供し、

上記制御プロトコルユニットは、制御プロトコルに固有の機能を提供することを特徴とする請求項１記載の機能モジュール。

【請求項３】　上記汎用ユニットは、論理デバイスマネージャを備え、該論理デバイスマネージャは、利用可能なホームネットワークデバイスのリストを維持し、該論理デバイスマネージャに参加可能なデバイスマネージャイベントリスナオブジェクトのリストを維持し、参加したデバイスマネージャイベントリスナオブジェクトに対し、ホームネットワーク機器に関するホームネットワークイベント情報の少なくとも一部をメッセージとして送る制御プロトコルから独立した機能を提供することを特徴とする請求項２記載の機能モジュール。

【請求項４】　上記汎用ユニットは、上記論理デバイスマネージャに接続され、上記ホームネットワーク機器の１つに対応する０以上の論理デバイスを備え、該論理デバイスは、該対応するホームネットワーク機器と通信を行い、該論理デバイスにおいて参加可能なデバイスメッセージリスナオブジェクトに対し、ホームネットワーク機器イベント情報の少なくとも一部をメッセージとして送るための制御プロトコルから独立した機能を提供することを特徴とする請求項３記載の機能モジュール。

【請求項５】　上記制御プロトコルユニットは、上記論理デバイスマネージャの全ての機能及び制御プロトコル固有の追加的機能を提供する制御プロトコルデバイスマネージャを備えることを特徴とする請求項３又は４記載の機能モジュール。

【請求項６】　上記制御プロトコルユニットは、上記少なくとも１つの論理デバイスの１つに対応し、該対応する論理デバイスの機能の他に、制御プロトコル固有の追加的機能を提供する０以上の制御プロトコルデバイスを備えることを特徴とする請求項４又は５記載の機能モジュール。

【請求項７】　上記制御プロトコルユニットは、上記制御プロトコルデバイスマネージャ及び上記適応化手段の間のデバイス独立インタフェース構造を提供する制御プロトコルデバイスクリエータインタフェースを備えることを特徴とする請求項５又は６記載の機能モジュール。

【請求項８】　上記適応化手段は、少なくとも１つのデバイスサブユニットクリエータオブジェクトクラスを備え、上記デバイス制御手段は、上記少なくとも１つの各デバイスサブユニットクリエータオブジェクトクラスにそれぞれ対応する少なくとも１つのデバイスサブユニット制御オブジェクトクラスを備え、上記少なくとも１つのデバイスサブユニット制御オブジェクトクラスに由来するオブジェクトは、上記制御プロトコルデバイスマネージャから上記制御プロトコルデバイスクリエータインタフェースを介して上記少なくとも１つのデバイスサブユニットクリエータオブジェクトクラスの１つに供給されるホームネットワークデバイスイベント情報に基づいて生成／適応可能であることを特徴とする請求項７記載の機能モジュール。

【請求項９】　上記デバイス制御手段は、制御ユニット開発者が上記デバイスサブユニットクリエータオブジェクトクラスを変更／拡張することにより、変更／拡張可能であることを特徴とする請求項１乃至８いずれか１項記載の機能モジュール。

【請求項１０】　上記適応化手段は、制御ユニット開発者が上記デバイスサブユニット制御オブジェクトクラスを変更／拡張することにより、変更／拡張可能であることを特徴とする請求項１乃至９いずれか１項記載の機能モジュール。

【請求項１１】　当該機能モジュールは、ソフトウェアモジュールとして実現されていることを特徴とする請求項１乃至１０いずれか１項記載の機能モジュール。

【請求項１２】　上記機能モジュールは、Ｊａｖａパッケージとして実現され、上記管理ユニットは、管理層として実現され、上記デバイスユニットは、デバイス層として実現されていることを特徴とする請求項１乃至１１いずれか１項記載の機能モジュール。

【請求項１３】　外部機能モジュールを制御サーバ及び情報提供サーバとして使用する制御ユニットによりホームネットワーク機器を制御するホームネットワーク機器制御方法において、

上記制御モジュールは、ホームネットワークにおけるホームネットワーク機器関連イベントを登録し、対応するホームネットワークイベント情報を解釈し、上記解釈されたホームネットワークイベント情報を用いて上記機能モジュールに含まれるデバイス制御手段を上記ホームネットワーク機器関連イベントの結果生じるホームネットワーク機器の状態の変化に適応化し、

上記制御ユニットは、上記適応化されたデバイス制御手段を用いて上記ホームネットワーク機器を制御することを特徴を特徴とするホームネットワーク機器制御方法。

【請求項14】 上記ホームネットワークイベント情報の少なくとも一部を上記機能モジュールから制御ユニットにメッセージとして送ることを特徴とする請求項13記載のホームネットワーク機器制御方法。

【請求項15】 上記制御ユニットを上記機能モジュールにおいてデバイスマネージャイベントリスナとして参加させ、該機能モジュールに対して、上記ホームネットワーク機器から該機能モジュールに送信されたホームネットワークイベント情報を該制御ユニットにメッセージとして送ることを特徴とする請求項14記載のホームネットワーク機器制御方法。

【請求項16】 上記制御ユニットを上記機能モジュールにおいてデバイスメッセージリスナとして参加させ、該機能モジュールに対して、上記ホームネットワーク機器から該機能モジュールに送信されたホームネットワークイベント情報を該制御ユニットにメッセージとして送ることを特徴とする請求項14又は15記載のホームネットワーク機器制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ホームネットワークを制御する機能モジュール及びホームネットワーク機器制御方法に関する。

【0002】

【従来の技術】Ｊａｖａ（登録商標）ホームネットワークアプリケーションを用いて、ホームネットワーク機器を制御する技術が知られている。この制御のために、通常、Ｊａｖａホームネットワークインタフェースを機能モジュールとして用いる。Ｊａｖａホームネットワークインタフェースは、Ｊａｖａアプリケーションに、ホームネットワーク機器を制御するのに必要な機能を提供する。例えば、ホームオーディオビジュアルインタオペラビリティ（home audio/visual interoperability：以下、ＨＡＶｉという。）仕様は、ホームネットワーク機器を制御するためのＪａｖａホームネットワークインタフェースとしてのＪａｖａアーキテクチャを定義し、これにより、ホームネットワークには、コンピュータシステムネットワークと共通の様々な機能が提供される。このようなＨＡＶｉアーキテクチャについては、例えば、欧州特許公開公報ＥＰ０９２９１７０Ａ２号に開示されている。

【0003】

【発明が解決しようとする課題】ＨＡＶｉアーキテクチャは、非常に複雑なソフトウェアホームネットワークインタフェースを用いるため、多くの演算リソースを必要とする。ＨＡＶｉアーキテクチャを使用するＪａｖａアプリケーションは、自らが提供する機能の一部を必要とするだけではなく、大きな「機能的オーバーヘッド（functionality overhead）」を「抱える（carrying）」という問題がある。

【0004】本発明は、上述の課題に鑑みてなされたものであり、本発明の目的は、単純な構成で演算リソースを節約することができる機能モジュール及びホームネットワーク機器制御方法を提供することである。

【0005】

【課題を解決するための手段】上述の目的を達成するために、本発明に係る機能モジュールは、ホームネットワーク機器の制御ユニットにより使用される機能モジュールにおいて、ホームネットワークにおけるホームネットワーク機器に関するイベントを登録し、イベントをホームネットワーク機器に送信する機能を提供する管理ユニットと、管理ユニットに接続され、適応化手段及びデバイス制御手段を有するデバイスユニットとを備える。デバイス制御手段は、ホームネットワーク機器固有の機能を提供し、ホームネットワーク機器を制御し、適応化手段により適応化されて、ホームネットワーク機器に関するイベントによりホームネットワーク機器の状態を変更する。

【0006】さらに、上述の目的を達成するために、本発明に係るホームネットワーク機器制御方法は、外部機能モジュールを制御サーバ及び情報提供サーバとして使用する制御ユニットによりホームネットワーク機器を制御するホームネットワーク機器制御方法において、制御モジュールは、ホームネットワークにおけるホームネットワーク機器関連イベントを登録し、対応するホームネットワークイベント情報を解釈し、上記解釈されたホームネットワークイベント情報を用いて機能モジュールに含まれるデバイス制御手段をホームネットワーク機器関連イベントの結果生じるホームネットワーク機器の状態の変化に適応化し、制御ユニットは、適応化されたデバイス制御手段を用いてホームネットワーク機器を制御する。

【0007】機能モジュール及び制御ユニットは、それぞれ、ハードウェアで構成してもよく、ソフトウェアで構成してもよく、これらを組み合わせて構成してもよい。好ましい実施の形態においては、機能モジュール及び制御ユニットをいずれもソフトウェアにより実現する。以下の説明では、制御ユニットをＪａｖａアプリケーションとして実現し、機能モジュールをＪａｖａパッケージとして実現する好ましい実施の形態を用いて、本発明の原理を説明する。さらに、管理ユニット及びデバイスユニットは、機能モジュールを表すＪａｖａパッケージ内の管理層及びデバイス層と呼ぶ。

【0008】本発明は、階層構造を有するオブジェクト指向の機能モジュールアーキテクチャ（具体例では、Ｊａｖａパッケージとして実現される）を実現し、ここで、アプリケーションにより無条件に必要とされる全て

の機能は、制御すべき機器の種類にかかわらず、下位層に設けられ、機器固有の機能は上位層に設けられる。このような構造により、一般的機能と特別な機能とを明確に分離することができ、したがって、Ｊａｖａパッケージの複雑性を減じることができる。さらに、このような構造により、Ｊａｖａパッケージは、下位層を変更することなく、上位層を変更／拡張することのみにより、容易に変更／拡張することができる。好ましくは、「コア」となる機能は、管理層に集中的に設けられ、管理層は、特に、Ｊａｖａアプリケーションとホームネットワーク機器との間の通信のための、機器から独立した機能を提供する。

【０００９】ホームネットワーク内に含まれる全ての機器の物理的存在及びこれらの機器の内部状態を全体的にホームネットワーク機器状態として表現することができる。例えば、オフに切り換えられているテレビジョンとオンに切り換えられているテープ装置のみからなるホームネットワークは、「テレビジョン／オフ、テープ／オン」として表現することができる。また、例えば、テープ装置をオフに切り換えられたチューナに置換した場合、ホームネットワーク機器状態は、「テレビジョン／オフ、チューナ／オフ」と表現することができる。管理層の機能は、実際のホームネットワーク機器状態又はこのホームネットワーク機器状態を変更するイベントを登録することであり、すなわち、管理層の機能は、ネットワーク内で「何が起きているか」に関する全ての情報を抽出することである。ホームネットワーク機器状態を変更するイベントは、Ｊａｖａパッケージにとって「外部イベント」であり、ホームネットワーク機器関連イベント（home network device correlated events）と呼ばれる。このようなホームネットワーク機器関連イベントは、対応するネットワーク機器固有の情報とともに、「ホームネットワークイベント情報」と呼ばれる。

【００１０】ホームネットワーク機器関連イベントの具体例を説明する。ホームネットワークに新たなホームネットワーク機器（以下、単に機器という。）が追加された場合、管理層は、ホームネットワーク機器関連イベント（以下、単に機器関連イベントという。）として、「機器追加」イベントを登録する。管理層は、この機器と通信を行い、機器固有の情報を抽出する。例えば、管理層は、新たな機器がステレオユニットであることを検出し、例えばチューナサブユニット、テープサブユニット、ディスクサブユニット等、このステレオユニットに含まれる全てのサブユニットに関する情報等、さらなる情報を抽出する。

【００１１】Ｊａｖａパッケージ自身の機能により開始されたイベント（ここでは、内部イベントと呼ぶ。）は、機器関連イベントとは区別する必要がある。例えば、Ｊａｖａパッケージの機能がテープ装置のテープが終了したこと（機器関連イベント）を認識した場合、Ｊａｖａパッケージの他の部分又はＪａｖａアプリケーションへの呼出を行うコールバック機能が実行される。コールバック機能の呼出の処理は、機器関連イベントとはみなされない内部イベントであるが、通常、機器関連イベントの結果として生じるイベントであり、したがって、機器関連イベントに割り当ててもよい。内部イベントに関する情報は、「内部イベント情報」とみなされる。

【００１２】デバイス層は、デバイス制御手段を備え、デバイス制御手段は、Ｊａｖａアプリケーションが機器を制御するために直接利用可能な機器固有の機能を提供する。例えば、デバイス制御手段は、チューナ装置を切り換え、又はテープ装置に挿入されているテープをイジェクトする等の機能（コマンド）を有している。デバイス制御手段の機能は、Ｊａｖａパッケージの全ての機能のうちで最も細分化された機能であり、すなわち、最も特化された機能である。

【００１３】さらに、本発明の具体例において、デバイス層は、デバイス制御手段自身を実際のホームネットワーク機器状態に適応化することができる。ホームネットワーク機器状態は、上述のように機器関連イベントにより変更可能である。したがって、デバイス層は、新たなデバイス制御手段機能を生成し、既存のデバイス制御手段機能を削除し、又は既存のデバイス制御手段機能を変更する能力を有する適応化手段を備える。適応化手段自身は、ユーザ又はＪａｖａアプリケーションにより拡張又は変更可能である。

【００１４】適応化手段は、管理層の機能により適用される実際のホームネットワーク機器状態に関する情報を用いる。すなわち、適応化手段は、状態情報又は単一のイベントの情報をホームネットワークイベント情報として解釈し、これに応じてデバイス制御手段を適応化する。これにより、ホームネットワークの状態の変化に関連するデバイス制御手段の変更は、完全にＪａｖａパッケージ自身が行うため、Ｊａｖａアプリケーションは、ホームネットワークの状態の変化を考慮する必要がない。例えば、ステレオユニットからチューナサブユニットが切り離された場合、管理層はこのイベントを機器関連イベントとして認識し、適応化手段に対応するホームネットワークイベント情報をメッセージとして送る。適応化手段は、このホームネットワークイベント情報を解釈し、例えば、デバイス制御手段内の対応するチューナ機能を削除することを決定する。これにより、Ｊａｖａアプリケーションには、以後チューナを制御することはできないことが通知される。また、新たな機器がホームネットワークに追加された場合、管理層は、このイベントを認識し、この機器から機器固有情報を読み出し、この機器固有情報をホームネットワークイベント情報として適応化手段に供給する。次に、適応化手段は、読み出された機器情報に対応する全ての機能を生成する。Ｊａ

ｖａアプリケーションは、デバイス制御手段の機能が変更され、その振舞がこれに応じて適応化されたことを認識する。

【００１５】好ましい実施の形態においては、管理層は、汎用層（general layer）及び制御プロトコル層を備え、汎用層は、制御プロトコル層及びデバイス層に共通の機能を提供し、制御プロトコル層は、制御プロトコル固有の機能を提供する。このように機能を明確に分離することにより、制御プロトコル機能が汎用層に含まれなくなるため、Ｊａｖａパッケージにより使用されている制御プロトコルを容易に変更することができる。制御プロトコルを変更するためには、制御プロトコル層の制御プロトコル固有の機能のみを変更すればよい。汎用層の機能は、制御プロトコルからは独立している。

【００１６】この原理は、制御プロトコル層及びデバイス層の間でも用いることができる。すなわち、制御プロトコル層は、機器固有の独立した機能のみを有する。すなわち、各上位層は、対応する下位層の少なくとも一部の機能と、さらなる機能とを含んでいる。

【００１７】換言すれば、下位層の原理（機能）は、全ての上位層で使用される。これにより、必要とする演算リソースが少なく、容易に拡張できる簡潔なアーキテクチャを実現することができる。

【００１８】ここでは、本発明の原理を説明するためにＪａｖａを用いているが、Ｃ＋＋やスモールトーク（SMALLTALK）等の他のいかなるプログラミング言語を機能モジュールの基礎として用いてソフトウェアパッケージを実現してもよい。さらに、上述のように、機能モジュールは、ハードウェア、又はハードウェアとソフトウェアの組合せにより実現してもよい。

【００１９】

【発明の実施の形態】以下、本発明に係る機能モジュール及びホームネットワーク機器制御方法について、図面を参照して詳細に説明する。

【００２０】図１は、本発明を適用したＪａｖａパッケージ１のアーキテクチャの具体例を示す図である。

【００２１】Ｊａｖａパッケージ１は、汎用層（general layer）２、制御プロトコル層３及びデバイス層４から構成されている。汎用層２は、ホームネットワーク内の利用可能な機器のリストを維持するプロトコルから独立した機能を提供する論理デバイスマネージャ（logical device manager）５を備え、すなわち、論理デバイスマネージャ５は、論理デバイスマネージャ５に参加可能なデバイスマネージャイベントリスナオブジェクト（manager event listener objects subscribable）のリストを維持し、参加しているデバイスマネージャイベントリスナオブジェクトに対し、デバイスイベント情報の少なくとも一部をメッセージとして送る。デバイスマネージャイベントリスナオブジェクトに対するメッセージの送信処理及びデバイスマネージャイベントリスナオブジェクトによる論理デバイスマネージャ５への参加処理は、デバイスマネージャイベントリスナインタフェース６により行われる。したがって、論理デバイスマネージャ５に参加を希望する全てのデバイスマネージャイベントリスナオブジェクトは、デバイスマネージャイベントリスナインタフェース６を実装している必要がある。論理デバイスマネージャ５は、好ましくは、ホームネットワークに新たな機器が接続され、或いは既存の機器が切り離されたことに関するイベントをデバイスマネージャイベントリスナオブジェクトにメッセージとして送る。

【００２２】汎用層２は、また、ホームネットワーク機器の数（０以上）に応じて、論理デバイスマネージャ５に接続された０以上の論理デバイス７を備え、各論理デバイス７は、それぞれホームネットワーク機器の１つに対応している。各論理デバイス７は、対応する物理デバイスユニットと通信を行い、及び論理デバイス７に参加可能なデバイスイベントリスナオブジェクトにホームネットワークイベント情報の一部をメッセージとして送るための制御プロトコルから独立した通信機能を提供する。例えば、テープデバイスにおいて駆動されているテープが端部に到達したというイベントは、Ｊａｖａアプリケーションに直接メッセージとして送られる。この意味で、Ｊａｖａパッケージ１は、情報クライアントにホームネットワークイベント情報を直接送信できる情報提供サーバとみなすことができる。なお、ここで、クライアントとは、デバイスメッセージリスナとなるＪａｖａアプリケーションである。

【００２３】デバイスメッセージリスナオブジェクトへのメッセージ送信処理及び論理デバイス７におけるデバイスメッセージリスナオブジェクトの参加処理は、デバイスメッセージリスナインタフェース８により実行される。したがって、論理デバイス７と通信することを望む全てのデバイスメッセージリスナオブジェクトは、デバイスメッセージリスナインタフェース８を実装している必要がある。

【００２４】デバイスマネージャイベントリスナオブジェクト及び／又はデバイスメッセージリスナオブジェクトは、Ｊａｖａパッケージ１における上位のソフトウェア層３、４のソフトウェアオブジェクトであってもよく、Ｊａｖａパッケージ１を用いたＪａｖａアプリケーションのソフトウェアオブジェクトであってもよく、ホームネットワーク内の各機器に割り当てられたソフトウェアオブジェクトであってもよい。

【００２５】制御プロトコル層３は、論理デバイスマネージャ５の全ての機能及び追加的な制御プロトコル固有の機能を実現する制御プロトコルデバイスマネージャ９を備える。さらに、制御プロトコル層３は、それぞれ論理デバイス７の１つに対応し、論理デバイス７の機能に加えて制御プロトコル固有の機能を追加する０以上の制御プロトコルデバイス１０を備える。制御プロトコル層

3は、さらに、制御プロトコルデバイスマネージャ９と、デバイス層４に含まれる適応化モジュール１２との間のデバイスから独立したインタフェース構造を提供する制御プロトコルデバイスクリエータインタフェース（control protocol device creator Interface）１１を備える。

【００２６】制御プロトコル固有の機能とは、例えば、オーディオ／ビデオ制御（Audio-Video/Control：以下、ＡＶ／Ｃという。）機能であってもよい。なお、制御プロトコル固有の機能は、他の適切ないかなる機能であってもよい。

【００２７】階層の機能は、矢印２０で示すように、継承（inheritance）により、上位層に転送される。これは、例えば、制御プロトコルデバイス１０であるＡＶ／Ｃデバイスの機能が論理デバイス７の機能を継承し、及びＡＶ／Ｃ固有の機能をこれに追加することにより生成されることを意味する。すなわち、ＡＶ／Ｃデバイスの機能は、例えばチューナ、ディスク再生装置等のＡＶ／Ｃデバイスが有するＡＶ／Ｃに基づくメッセージの送受信等、全てのＡＶ／Ｃデバイスに共通の機能を含んでいる。

【００２８】好ましい具体例においては、全ての種類の制御プロトコルデバイス１０に対して、対応する制御プロトコルデバイスクリエータインタフェース１１が設けられている。

【００２９】適応化モジュール１２は、例えば、３つのサブユニットクリエータ、すなわちデバイスサブユニットタイプ１クリエータ１３、デバイスサブユニットタイプ２クリエータ１４、デバイスサブユニットタイプ３クリエータ１５を備える。デバイス層４は、さらに、０以上のデバイスサブユニットコントローラを備えるデバイス制御モジュール１６を備え、ここでデバイスサブユニットタイプ１コントローラ１７、デバイスサブユニットタイプ２コントローラ１８、デバイスサブユニットタイプ３コントローラ１９は、それぞれデバイスサブユニットクリエータ１３、１４、１５の１つに対応している。デバイスサブユニットクリエータ１３、１４、１５及びデバイスサブユニットコントローラ１７、１８、１９は、それぞれオブジェクトクラスの形式で実現されている。デバイスサブユニットクリエータオブジェクトクラスに由来する各オブジェクトは、対応するオブジェクトデバイスサブユニットコントローラオブジェクトクラスを生成することができる。例えば、２つの同一の物理的ＡＶ／Ｃチューナ装置がホームネットワークに接続された場合、対応するＡＶ／Ｃチューナクリエータオブジェクトは、同じＡＶ／Ｃチューナコントローラオブジェクトクラスに由来する２つのＡＶ／Ｖチューナコントローラオブジェクトを生成する。この結果、各個別の物理デバイスユニットは、自らのデバイスサブユニットコントローラオブジェクトに対応し、物理的デバイスの各機能タイプは、自らのデバイスサブユニットクリエータオブジェクト及び自らのデバイスサブユニットコントローラオブジェクトクラスに対応する。

【００３０】このように、デバイスサブユニットコントローラ１７、１８、１９に由来する各オブジェクトは、ホームネットワークイベント情報に基づき、制御プロトコルデバイスクリエータインタフェース１１を介して、制御プロトコルデバイスマネージャ９からデバイスサブユニットクリエータ１３、１４、１５に供給可能な情報に基づき、対応するデバイスサブユニットクリエータ１３、１４、１５に由来するオブジェクトにより生成及び適応可能である。デバイスサブユニットコントローラ１７、１８、１９の機能は、制御プロトコルデバイス１０の機能から導出することができ、実際のデバイスを制御するために必要な特別な機能を追加的に含んでいる。例えば、デバイスサブユニットコントローラ１７、１８、１９は、それぞれ、ＡＶ／Ｃチューナコントローラ、ＡＶ／Ｃディスクコントローラ、ＡＶ／Ｃテープコントローラであってもよい。これに対応して、デバイスサブユニットクリエータ１３、１４、１５は、それぞれＡＶ／Ｃチューナクリエータ、ＡＶ／Ｃディスククリエータ、ＡＶ／Ｃテープクリエータであってもよい。

【００３１】制御プロトコルデバイスマネージャ９は、制御プロトコルデバイスクリエータインタフェース１１の実装（implementation）を用いることにより、デバイスサブユニットコントローラ１７、１８、１９に由来するオブジェクトの生成又は適応を開始する。制御プロトコルデバイスマネージャ９は、ホームネットワーク内の機器から、ホームネットワークイベント情報として、デバイスユニット情報又はデバイスサブユニット情報を読み出し、好ましくは自らデバイスユニット情報を解釈し、デバイスユニットに対応する制御プロトコルデバイスクリエータインタフェース１１を介して、適応化モジュール１２にデバイスサブユニット情報の解釈を指示する。

【００３２】適応化モジュール１２及び／又は制御モジュール１６の機能は、Ｊａｖａアプリケーション開発者により拡張及び変更可能である。このような拡張及び変更は、適応化モジュール１２において、デバイスサブユニットクリエータオブジェクトクラスを追加／削除／変更し、及び制御モジュール１６において、対応するデバイスサブユニットコントローラオブジェクトクラスを追加／削除／変更することにより実行される。

【００３３】論理デバイスマネージャ５、論理デバイス７、制御プロトコルデバイスマネージャ９、制御プロトコルデバイス１０、制御プロトコルデバイスクリエータインタフェース１１、適応化モジュール１２、デバイスサブユニットクリエータ１３、１４、１５、デバイス制御モジュール１６及びデバイスサブユニットコントローラ１７、１８、１９の機能は、それぞれＪａｖａクラス

により実現される。

【００３４】この具体例に示すＪａｖａパッケージアーキテクチャの利点は、拡張が容易であるという点にある。新たなデバイスタイプをサポートする必要が生じた場合、適応化モジュール１２内に、この新たなデバイスに関する情報を含む対応する新たなサブユニットクリエータオブジェクトクラスを実現するだけでよい。さらに、デバイス制御モジュール１６の機能は、対応するデバイスサブユニット制御オブジェクトクラスにより拡張することができる。アーキテクチャ内の他の部分に変更を加える必要はない。

【００３５】以下、論理デバイスマネージャ５、論理デバイス７、制御プロトコルデバイスマネージャ９、制御プロトコルデバイス１０、制御プロトコルデバイスクリエータインタフェース１１、適応化モジュール１２、デバイスサブユニットクリエータ１３、１４、１５、デバイス制御モジュール１６及びデバイスサブユニットコントローラ１７、１８、１９の可能な機能の具体例について、アプリケーションプログラミングインタフェース（application programming Interface：以下、ＡＰＩという。）の形式で示す。この具体例では、制御プロトコルとして、ＡＶ／Ｃプロトコルを使用している。

【００３６】

【表１】

表１　論理デバイスマネージャＡＰＩのリスト

ＡＰＩ機能	説明
AVCDeviceManager:: RegisterAVCDeviceCreator	ある種のＡＶ／Ｃデバイスを生成するオブジェクトのリファレンスを渡すことにより、ＡＶ／Ｃクリエータとして登録する。
AVCDeviceManager:: UnregisterAVCDeviceCreator	ＡＶ／Ｃデバイスデバイスクリエータとしての登録を抹消することを望むオブジェクトのリファレンスを渡すことにより、登録を抹消する。
AVCDeviceCreator::CreateDevice	ＡＶ／Ｃデバイスクリエータは、この機能を実装しなくてはならない。この機能は、ＡＶ／Ｃデバイスマネージャがデバイスの生成を開始するために呼び出される。パラメータとして、サブユニットタイプが渡される。
LogicalDeviceManager::GetDeviceList	論理デバイスオブジェクトのリストを返す。
LogicalDeviceManager::Subscribe	通知を希望するオブジェクトのリファレンスを渡すことにより、デバイスマネージャイベントリスナとして参加する。このオブジェクトは、デバイスイベントリスナインタフェースを実装する必要がある。
LogicalDeviceManager::Unsubscribe	以後通知は不要であることを示すオブジェクトのリファレンスを渡すことにより、デバイスマネージャイベントリスナとしての参加を取り消す。
EventManagerEventListener:: NewDevice	論理デバイスマネージャにより呼び出され、参加オブジェクトに対し、ネットワークに追加されたデバイスに関する情報を通知する。
EventManagerEventListener:: GoneDevice	論理デバイスマネージャにより呼び出され、参加オブジェクトに対し、ネットワークから削除されたデバイスに関する情報を通知する。

【００３７】

【表２】

表2　論理デバイスＡＰＩのリスト

ＡＰＩ機能	説明
LogicalDevice::GetGuid	論理デバイスオブジェクトのＧＵＩＤを返す。
LogicalDevice::Subscribe	通知を希望するオブジェクトのリファレンスを渡すことにより、デバイスメッセージリスナとして参加する。このオブジェクトは、デバイスメッセージリスナインタフェースを実装する必要がある。
LogicalDevice::Unsubscribe	以後通知は不要であることを示すオブジェクトのリファレンスを渡すことにより、デバイスメッセージリスナとしての参加を取り消す。
LogicalDevice::Send	ＦＣＰメッセージを送信する必要がある場合にこの機能を使用する。
DeviceMessageListener::Receive	デバイスオブジェクトにより呼び出され、参加オブジェクトに対し、デバイスオブジェクトにより表されているデバイスから到着したメッセージを通知する。

【００３８】　【表３】

表３　ＡＶ／Ｃデバイス ＡＰＩリスト

ＡＰＩ機能	説明
AVCDevice::GetAVCDeviceType	デバイスタイプ（チューナ、ディスク等）を返す。
AVCDevice::Send	ＡＶ／Ｃメッセージを送信する必要がある場合にこの機能を使用する。

【００３９】　【表４】

表４　ＡＶ／Ｃテープ ＡＰＩリスト

ＡＰＩ機能	説明
AVCTape::Play	挿入されたテープの再生を開始する。
AVCTape::Stop	テープの全ての動作を停止する。
AVCTape::Record	挿入されたテープにデータを記録する。
AVCTape::EjectMedium	テープをイジェクトする。
AVCTape::FastRewind	テープを巻き戻す。
AVCTape::FastForward	テープを早送りする。
AVCTape::Pause	現在の動作を一時停止する。

【００４０】以下、図２を参照して、機能モジュールとしてのＪａｖａパッケージと、制御ユニットとしてのＪａｖａアプリケーションと、ネットワークソフトウェアとを含む汎用ソフトウェアアーキテクチャ（general software architecture）の具体例を説明する。

【００４１】汎用ソフトウェアアーキテクチャ３０は、例えば、ｉ．ＬＩＮＫ又はＩＥＥＥ１３９４インタフェース等、ホームネットワークバス２１を介して通信を行うためのネイティブホームネットワークドライバインタフェース２２を備える。ネイティブホームネットワークドライバインタフェース２２は、ホストオペレーティングシステム２３により制御される。ホストオペレーティングシステム２３は、自ら、Ｊａｖａランタイム環境（仮想マシン）の基礎として機能する。例えば上述したようなＪａｖａパッケージ２５は、自らのタスクを実行するために、Ｊａｖａランタイム環境２４を使用し、常駐Ｊａｖａアプリケーション２６又はダウンロードされたＪａｖａアプリケーション２７は、Ｊａｖａパッケージ２５を用いて、ユーザ２８によりＪａｖａアプリケーション２６、２７に与えられたタスクを実行する。

【００４２】本発明は、好ましくは、基底に存在するハードウェアを抽象化するＪａｖａパッケージとして実現

される機能モジュールを提供し、これにより、複雑なユーザインタフェースを用いることなく、ホームネットワーク機器に情報を提供し、及びホームネットワーク機器を管理することができる。すなわち、本発明によれば、アプリケーションは、柔軟にホームネットワーク機器を制御することができる。

【００４３】なお、機能モジュール及び制御ユニットは、それぞれ、ハードウェアで構成してもよく、ソフトウェアで構成してもよく、これらを組み合わせて構成してもよい。上述の実施例においては、機能モジュール及び制御ユニットをいずれもソフトウェアにより実現している。

【００４４】以上の説明でも明らかなように、本発明では、階層構造を有するオブジェクト指向の機能モジュールアーキテクチャ（具体例では、Ｊａｖａパッケージとして実現される）を実現し、ここで、アプリケーションにより無条件に必要とされる全ての機能は、制御すべき機器の種類にかかわらず、下位層に設けられ、機器固有の機能は上位層に設けられる。このような構造により、一般的機能と特別な機能とを明確に分離することができ、したがって、Ｊａｖａパッケージの複雑性を減じることができる。さらに、このような構造により、Ｊａｖａパッケージは、下位層を変更することなく、上位層を変更／拡張することのみにより、容易に変更／拡張することができる。好ましくは、「コア」となる機能は、管理層に集中的に設けられ、管理層は、特に、Ｊａｖａアプリケーションとホームネットワーク機器との間の通信のための、機器から独立した機能を提供する。

【００４５】ホームネットワーク内に含まれる全ての機器の物理的存在及びこれらの機器の内部状態を全体的にホームネットワーク機器状態として表現することができる。例えば、オフに切り換えられているテレビジョンとオンに切り換えられているテープ装置のみからなるホームネットワークは、「テレビジョン／オフ、テープ／オン」として表現することができる。また、例えば、テープ装置をオフに切り換えられたチューナに置換した場合、ホームネットワーク機器状態は、「テレビジョン／オフ、チューナ／オフ」と表現することができる。管理層の機能は、実際のホームネットワーク機器状態又はこのホームネットワーク機器状態を変更するイベントを登録することであり、すなわち、管理層の機能は、ネットワーク内で「何が起きているか」に関する全ての情報を抽出することである。ホームネットワーク機器状態を変更するイベントは、Ｊａｖａパッケージにとって「外部イベント」であり、ホームネットワーク機器関連イベント（home network device correlated events）と呼ばれる。このようなホームネットワーク機器関連イベントは、対応するネットワーク機器固有の情報とともに、「ホームネットワークイベント情報」と呼ばれる。

【００４６】ホームネットワーク機器関連イベントの具体例を説明する。ホームネットワークに新たなホームネットワーク機器（以下、単に機器という。）が追加された場合、管理層は、ホームネットワーク機器関連イベント（以下、単に機器関連イベントという。）として、「機器追加」イベントを登録する。管理層は、この機器と通信を行い、機器固有の情報を抽出する。例えば、管理層は、新たな機器がステレオユニットであることを検出し、例えばチューナサブユニット、テープサブユニット、ディスクサブユニット等、このステレオユニットに含まれる全てのサブユニットに関する情報等、さらなる情報を抽出する。

【００４７】Ｊａｖａパッケージ自身の機能により開始されたイベント（ここでは、内部イベントと呼ぶ。）は、機器関連イベントとは区別する必要がある。例えば、Ｊａｖａパッケージの機能がテープ装置のテープが終了したこと（機器関連イベント）を認識した場合、Ｊａｖａパッケージの他の部分又はＪａｖａアプリケーションへの呼出を行うコールバック機能が実行される。コールバック機能の呼出の処理は、機器関連イベントとはみなされない内部イベントであるが、通常、機器関連イベントの結果として生じるイベントであり、したがって、機器関連イベントに割り当ててもよい。内部イベントに関する情報は、「内部イベント情報」とみなされる。

【００４８】デバイス層は、デバイス制御手段を備え、デバイス制御手段は、Ｊａｖａアプリケーションが機器を制御するために直接利用可能な機器固有の機能を提供する。例えば、デバイス制御手段は、チューナ装置を切り換え、又はテープ装置に挿入されているテープをイジェクトする等の機能（コマンド）を有している。デバイス制御手段の機能は、Ｊａｖａパッケージの全ての機能のうちで最も細分化された機能であり、すなわち、最も特化された機能である。

【００４９】さらに、本発明では、デバイス層は、デバイス制御手段自身を実際のホームネットワーク機器状態に適応化することができる。ホームネットワーク機器状態は、上述のように機器関連イベントにより変更可能である。したがって、デバイス層は、新たなデバイス制御手段機能を生成し、既存のデバイス制御手段機能を削除し、又は既存のデバイス制御手段機能を変更する能力を有する適応化手段を備える。適応化手段自身は、ユーザ又はＪａｖａアプリケーションにより拡張又は変更可能である。

【００５０】適応化手段は、管理層の機能により適用される実際のホームネットワーク機器状態に関する情報を用いる。すなわち、適応化手段は、状態情報又は単一のイベントの情報をホームネットワークイベント情報として解釈し、これに応じてデバイス制御手段を適応化する。これにより、ホームネットワークの状態の変化に関連するデバイス制御手段の変更は、完全にＪａｖａパッ

ケージ自身が行うため、Ｊａｖａアプリケーションは、ホームネットワークの状態の変化を考慮する必要がない。例えば、ステレオユニットからチューナサブユニットが切り離された場合、管理層はこのイベントを機器関連イベントとして認識し、適応化手段に対応するホームネットワークイベント情報をメッセージとして送る。適応化手段は、このホームネットワークイベント情報を解釈し、例えば、デバイス制御手段内の対応するチューナ機能を削除することを決定する。これにより、Ｊａｖａアプリケーションには、以後チューナを制御することはできないことが通知される。また、新たな機器がホームネットワークに追加された場合、管理層は、このイベントを認識し、この機器から機器固有情報を読み出し、この機器固有情報をホームネットワークイベント情報として適応化手段に供給する。次に、適応化手段は、読み出された機器情報に対応する全ての機能を生成する。Ｊａｖａアプリケーションは、デバイス制御手段の機能が変更され、その振舞がこれに応じて適応化されたことを認識する。

【００５１】上述の実施例では、管理層は、汎用層（general layer）及び制御プロトコル層を備え、汎用層は、制御プロトコル層及びデバイス層に共通の機能を提供し、制御プロトコル層は、制御プロトコル固有の機能を提供する。このように機能を明確に分離することにより、制御プロトコル機能が汎用層に含まれなくなるため、Ｊａｖａパッケージにより使用されている制御プロトコルを容易に変更することができる。制御プロトコルを変更するためには、制御プロトコル層の制御プロトコル固有の機能のみを変更すればよい。汎用層の機能は、制御プロトコルからは独立している。

【００５２】この原理は、制御プロトコル層及びデバイス層の間でも用いることができる。すなわち、制御プロトコル層は、機器固有の独立した機能のみを有する。すなわち、各上位層は、対応する下位層の少なくとも一部の機能と、さらなる機能とを含んでいる。

【００５３】換言すれば、下位層の原理（機能）は、全ての上位層で使用される。これにより、必要とする演算リソースが少なく、容易に拡張できる簡潔なアーキテクチャを実現することができる。

【００５４】なお、上述では、本発明の原理を説明するためにＪａｖａを用いているが、Ｃ＋＋やスモールトーク（SMALLTALK）等の他のいかなるプログラミング言語を機能モジュールの基礎として用いてソフトウェアパッケージを実現してもよい。さらに、上述のように、機能モジュールは、ハードウェア、又はハードウェアとソフトウェアの組合せにより実現してもよい。

【００５５】

【発明の効果】以上のように、本発明に係る機能モジュールは、ホームネットワークにおけるホームネットワーク機器に関するイベントを登録し、イベントをホームネットワーク機器に送信する機能を提供する管理ユニットと、管理ユニットに接続され、適応化手段及びデバイス制御手段を有するデバイスユニットとを備え、デバイス制御手段は、ホームネットワーク機器固有の機能を提供し、ホームネットワーク機器を制御し、適応化手段により適応化されて、ホームネットワーク機器に関するイベントによりホームネットワーク機器の状態を変更する。これにより、必要とする演算リソースが少なく、容易に拡張できる簡潔なアーキテクチャを実現することができる。

【図面の簡単な説明】

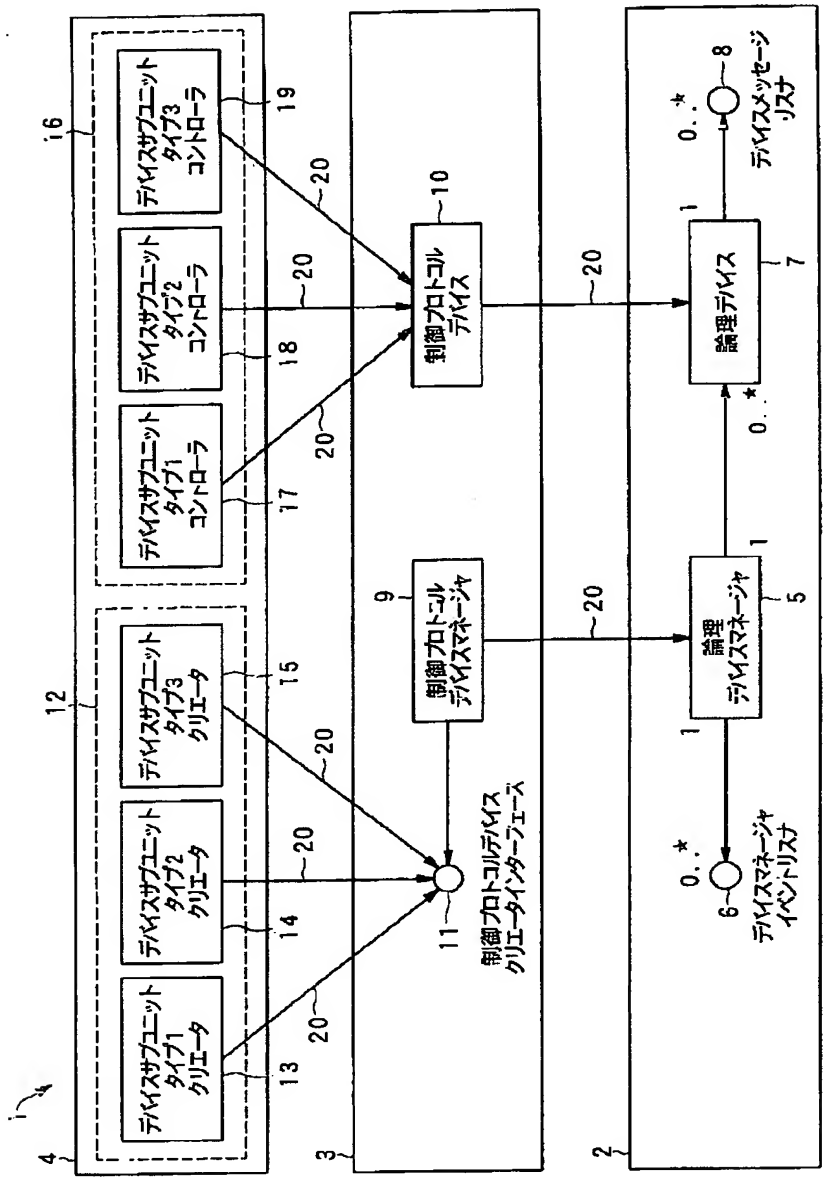
【図１】本発明に基づく３つの階層を有するＪａｖａパッケージの構造を示す図である。

【図２】本発明に基づくＪａｖａパッケージ、Ｊａｖａアプリケーション及びネットワークソフトウェアを含む汎用ソフトウェアアーキテクチャを示す図である。

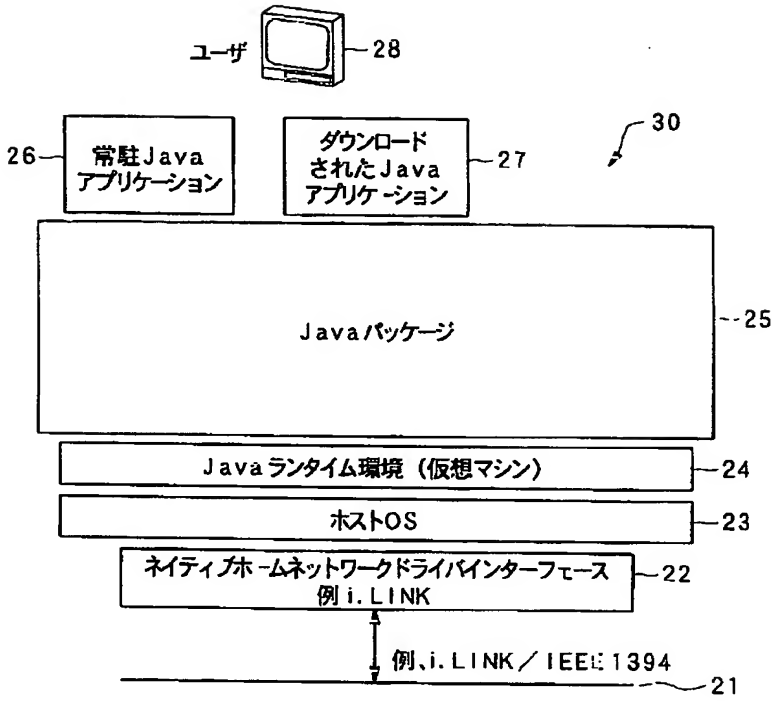
【符号の説明】

１　Ｊａｖａパッケージ、２　汎用層、３　制御プロトコル層、４　デバイス層、５　論理デバイスマネージャ、６　デバイスマネージャイベントリスナインタフェース、７　論理デバイス、８　デバイスメッセージリスナインタフェース、９　制御プロトコルデバイスマネージャ、１０　制御プロトコルデバイス、１１　制御プロトコルデバイスクリエータインタフェース、１２　適応化モジュール、１３　デバイスタイプ１クリエータ、１４　デバイスタイプ２クリエータ、１５　デバイスタイプ３クリエータ、１６　デバイス制御モジュール、１７　デバイスサブユニットタイプ１コントローラ、１８　デバイスサブユニットタイプ２コントローラ、１９　デバイスサブユニットタイプ３コントローラ

【図1】


デバイスサブユニット タイプ1 クリエータ
デバイスサブユニット タイプ2 クリエータ
デバイスサブユニット タイプ3 クリエータ
デバイスサブユニット タイプ1 コントローラ
デバイスサブユニット タイプ2 コントローラ
デバイスサブユニット タイプ3 コントローラ
制御プロトコル デバイスマネージャ
制御プロトコル デバイス
制御プロトコルデバイス クリエータインターフェース
論理デバイスマネージャ
論理デバイス
デバイスマネージャ イベントリスナ
デバイスメッセージ リスナ
0..*
1
4
12
16
13
14
15
17
18
19
20
3
9
10
11
2
5
7
6
8

【図2】



フロントページの続き

(72)発明者 シュッツ、 ポール
ドイツ連邦共和国 70327 シュトゥットゥガルト ハインリッヒーヘルツーシュトラーセ 1 ソニー インターナショナル（ヨーロッパ） ゲゼルシャフト ミット ベシュレンクテル ハフツング アドバンスド テクノロジー センター シュトゥットゥガルト内

(72)発明者 マイエル、 マッチアス
ドイツ連邦共和国 70327 シュトゥットゥガルト ハインリッヒーヘルツーシュトラーセ 1 ソニー インターナショナル（ヨーロッパ） ゲゼルシャフト ミット ベシュレンクテル ハフツング アドバンスド テクノロジー センター シュトゥットゥガルト内

(72)発明者 ティーデマン、 ステッフェン
ドイツ連邦共和国 70327 シュトゥットゥガルト ハインリッヒーヘルツーシュトラーセ 1 ソニー インターナショナル（ヨーロッパ） ゲゼルシャフト ミット ベシュレンクテル ハフツング アドバンスド テクノロジー センター シュトゥットゥガルト内

(72)発明者 テッラノヴァ、 サビーネ
ドイツ連邦共和国 70327 シュトゥットゥガルト ハインリッヒーヘルツーシュトラーセ 1 ソニー インターナショナル（ヨーロッパ） ゲゼルシャフト ミット ベシュレンクテル ハフツング アドバンスド テクノロジー センター シュトゥットゥガルト内

Fターム(参考) 5B089 GB01 GB08 HB11 JA35 KA11
5K033 BA01 BA08 CB14 DA01